虚拟机以及Dune的花活

对于计算机的一种模拟

- Guest OS【Guest 空间】（一个或者多个OS内核）（OS从内核上升到用户空间）（内核之上的container）
  - User Mode
  - Supervisor Mode
- Virtual Machine Monitor(VMM)【Host 空间】
  - 对于计算机的模拟（当然完全模拟会有点困难，出于性能）
- 硬件

云计算，低强度服务集成在一个物理机上，开发内核，用VMM抽象实现更多的功能（比如快照增加可靠性，支持多次运行，迁移到其他计算机上）安全性。虚拟机，逃逸，更加严格的隔离。

基本思路 trap and emulate

- 因为纯软件的模拟非常的慢，所以一种比较广泛的是在CPU上用Guest指令，把指令放到内存里面，然后让硬件直接跑。通过特殊的privileged指令产生的trap来追踪对应的信息。（ecall和sret也是特权指令，所以跳转是用户空间，ecall到VMM，然后处理对应系统调用，然后sret到VMM，然后设置对应guest为User Mode，然后回去。）
- 我们用VMM的trap代替了kernel的trap。当然guest的trap并不会实际设置SATP。因为VMM需要用真的寄存器所以我们就不能让Guest用真的寄存器了（会暂时借用VMM真实的寄存器）
- 普通指令硬件速度，所有涉及到kernel的应该会慢很多。
- 总之就是把不信任的guest kernel 变成自己信任的VMM。你可以把VMM运行在Smode（你可以裸机跑，也可以放在linux内核里面。）【不严格的模拟让linux还是知道自己运行在虚拟机上的，有时甚至可以利用这个特性来speed up】

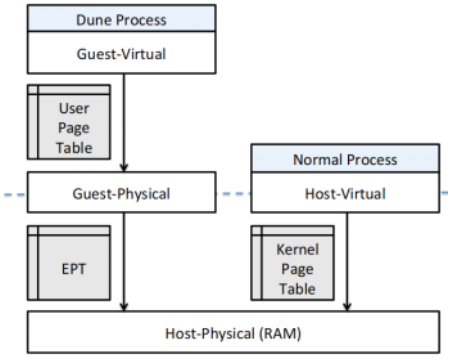
简述关于Guest的特权指令的处理

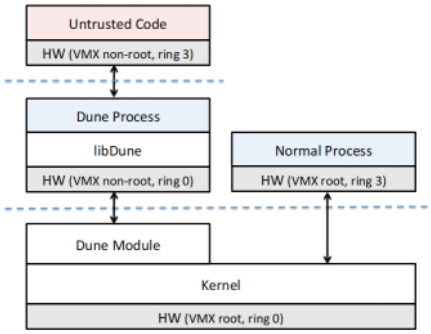
Pagetable 和外部设备

所以这里的目标是Guest可以在不触发trap的前提下，执行privileged指令。

我们还是有一个VMM在内核空间，并且Guest运行在用户空间。当我们使用这种新的硬件支持的方案时，我们的VMM会使用真实的控制寄存器，而当VMM通知硬件切换到Guest mode时，硬件里还会有一套完全独立，专门为Guest mode下使用的虚拟控制寄存器。在Guest mode下可以直接读写控制寄存器，但是读写的是寄存器保存在硬件中的拷贝，而不是真实的寄存器

实际的OS知道自己是不是运行在虚拟机里面的。





所以，当Guest执行sret指令从Supervisor mode进入到User mode，因为sret是privileged指令，会通过trap进入到VMM。VMM会更新虚拟状态信息中的mode为User mode，尽管当前的真实mode还是Supervisor mode，因为我们还在执行VMM中的代码。在VMM从trap中返回之前，VMM会将真实的SEPC寄存器设置成自己保存在虚拟状态信息中的虚拟SEPC寄存器。因为当VMM使用自己的sret指令返回到Guest时，它需要将真实的程序计数器设置成Guest操作系统想要的程序计数器值（注，因为稍后Guest代码会在硬件上执行，因此依赖硬件上的程序计数器）。所以在一个非常短的时间内，真实的SEPC寄存器与虚拟的SEPC寄存器值是一样的。同时，当VMM返回到虚拟机时，还需要切换Page table，这个我们稍后会介绍。

Guest中的用户代码，如果是普通的指令，就直接在硬件上执行。当Guest中的用户代码需要执行系统调用时，会通过执行ECALL指令（注，详见6.3，6.4）触发trap，而这个trap会走到VMM中（注，因为ECALL也是个privileged指令）。VMM可以发现当前在虚拟状态信息中记录的mode是User mode，并且发现当前执行的指令是ECALL，之后VMM会更新虚拟状态信息以模拟一个真实的系统调用的trap状态。比如说，它将设置虚拟的SEPC为ECALL指令所在的程序地址（注，执行sret指令时，会将程序计数器的值设置为SEPC寄存器的值。这样，当Guest执行sret指令时，可以从虚拟的SEPC中读到正确的值）；将虚拟的mode更新成Supervisor；将虚拟的SCAUSE设置为系统调用；将真实的SEPC设置成虚拟的STVEC寄存器（注，STVEC保存的是trap函数的地址，将真实的SEPC设置成STVEC这样当VMM执行sret指令返回到Guest时，可以返回到Guest的trap handler。Guest执行系统调用以为自己通过trap走到了Guest内核，但是实际上却走到了VMM，这时VMM需要做一些处理，让Guest以及之后Guest的所有privileged指令都看起来好像是Guest真的走到了Guest内核）；之后调用sret指令跳转到Guest操作系统的trap handler，也就是STVEC指向的地址。